

### **REMARKS/ARGUMENTS**

Favorable consideration of this application is respectfully requested.

Claims 1, 4, 6, 7, 11-15, and 18-20 are pending in this application. Claims 2, 3, 5, 8-10, 16, and 17 have been previously canceled without prejudice or disclaimer.

The outstanding Official Action presents a rejection of Claims 1, 4, 6, and 19 under 35 U.S.C. §103(a) as being unpatentable over Belu (U.S. Published Patent Application No. 2002/0033762, Belu '762) in view of Videcrantz et al. (U.S. Patent No. 6,275,588, Videcrantz), a rejection of Claims 7 and 11 under 35 U.S.C. §103(a) as being unpatentable over Belu (U.S. Published Patent Application No. 2002/0143792, Belu '792) in view of Videcrantz and "Winzip" (hereinafter "Winzip" screenshot documents), an improper rejection of Claims 12 and 13 lacking a citation of references relied upon or any stated statutory basis, and a rejection of Claims 13-15, 18, and 20 under 35 U.S.C. §103(a) as being unpatentable over Belu '762 in view of Videcrantz, "Winzip" screenshot documents, and Belu '792.

### **WITHDRAWAL OF IMPROPER REJECTIONS**

As the statement of the outstanding rejection of Claims 1, 4, 6, and 19 under 35 U.S.C. §103(a) as being unpatentable over Belu '762 in view of Videcrantz omits mention of reliance on the "Winzip" screenshot documents in violation of MPEP § 706.02(j) guidelines and *In re Hoch*, 428 F.2d 1341, 1342 n.3 166 USPQ 406, 407 n. 3 (CCPA 1970), withdrawal of this improperly stated rejection is clearly required.

The rejection of "claims 12 and 13 . . . under the same reasoning as claims 1 and 7" at lines 9 and 10 on page 6 of the outstanding Action also violates MPEP § 706.02(j) guidelines and the *Hoch* decision as no statement of the references relied upon appears. The further omission of any statutory basis for this rejection of Claims 12 and 13 also violates

MPEP § 707.07(d) and its requirement for a designation of the “*statutory basis* for any ground of rejection by express reference to a section of 35 U.S.C. in the opening sentence of each ground of rejection.”

As the requirements of the MPEP have not been followed as to providing a proper opening sentence statement of a rejection as to this rejection of Claims 12 and 13, withdrawal of this improperly stated rejection is also clearly required.

### **TRAVERSALS OF OUTSTANDING REJECTIONS**

In addition, the rejection of Claims 1, 4, 6, and 19 under 35 U.S.C. §103(a) as being unpatentable over Belu ‘762 in view of Videcrantzis with or without consideration of the “Winzip” screenshot documents omits consideration of all the limitations of base independent Claims 1 and 6, which limitations are incorporated into Claim 4 that depends on Claim 1 and Claim 19 that depends on Claim 6.

The Supreme Court decision of *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966) establishes that it is a first required to correctly analyze the claims to ascertain the differences between the claimed invention and the prior art. Thus, in order to establish a valid *prima facie* case of obviousness, the PTO must accurately set forth the subject matter claimed and the reasons why the artisan would have considered that actually claimed subject matter to have been obvious at the time the invention was made.

Here the requirement of base independent Claims 1 and 6 that the plurality of individual programs being compressed must “each include a same instruction set” is not adequately addressed by the outstanding Action that simply notes that paragraph [0051] of Belu ‘762 teaches that “[f]iles with the “.exe” extension are placed next to files with the “.dll” extension because they are determined to have the same ‘binary program’ file type.” The bottom of page 2 of the outstanding Action then concentrates on what a “binary file” is under

the “Encarta” definition followed by the “Encarta” definition of “machine language,” a term that only mentions instructions, and not any “instruction set,” the language in base independent Claims 1 and 6.

In this last regard, the claim language (“instruction set”) is a term widely used by those of ordinary skill in the art that has an ordinary and customary meaning with regard to the relationship of the “instruction set” to system architecture. Thus, while “instruction sets” certainly all include computer instructions, each individual instruction set includes computer instructions in a format required as to that “instruction set” architecture. Note the following from page 8 of the USPTO website version of U.S. Patent Number 6,219,774 (copy of page 8 attached):

One aspect of the invention is that the processor supports multiple system architectures. Thus, the *number of instruction sets* and/or system architectures supported, as well as the *type of instruction sets* and system architectures supported, are not critical to this aspect of the invention. What is important to this aspect of the invention is that the *processor can switch between the instruction set architectures and system architectures*. For example, alternative embodiments may support one instruction set and two system architectures. As another example, alternative embodiments may support three instruction set architectures and two system architectures. Other alternative embodiments may support three instruction set architectures and three system architectures. An embodiment which supports two instruction set architectures and two system architectures is described so as not to obscure the invention.

FIG. 2 shows a functional block diagram illustrating the selectable configurations or modes of a processor according to one embodiment of the invention. FIG. 2 shows a line 200 representing that the processor includes an instruction set unit 203 and a system unit 207. FIG. 2 also shows that instruction set unit 203 selectively *operates in either an instruction set configuration 210 or in an instruction set configuration 220*. In one embodiment, instruction set configuration 210 includes segmentation unit 215. Segmentation unit 215 allows for compatibility with existing x86 memory management techniques which utilize segmentation. In addition, FIG. 2 shows system unit 207, which selectively operates in either a system configuration 230 or a system configuration 240.

Instruction set unit 203 executes instructions from a first instruction set while instruction set configuration 210 is selected. In one embodiment, *this first instruction set is based on the 16/32-bit x86 instruction set used by existing Intel microprocessors*. This *instruction set operates using what are*

***referred to as effective or logical addresses.*** Instruction set configuration 210 sends these effective addresses to segmentation unit 215 which translates them into linear addresses. The technique of segmentation is well known in the prior art and is further described in the following reference: Shanley, Tom and Anderson, Don, ISA System Configuration, MindShare, Inc. (1993). Thus, instruction set configuration 210 with segmentation unit 215 provides a first instruction set architecture. Alternative embodiments which support ***other instruction sets may require other address translation techniques (rather than or in addition to segmentation), or may not require any address translation.***

Instruction set unit 203 executes instructions from ***a second instruction set which is different from the first instruction set***, while instruction set configuration 220 is selected. In one embodiment, this ***second instruction set is a 64-bit instruction set which operates using the same format of address generated by segmentation unit 215 (i.e., linear addresses).*** ***Since this 64-bit instruction set uses linear addresses, it can address the entire 64-bit virtual address space and does not require segmentation.*** In this manner, instruction set configuration 220 provides a second instruction set architecture. [Emphasis added.]

It is well established that "[t]he ordinary and customary meaning of a claim term is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention, *i.e.*, as of the effective filing date of the patent application." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1313, 75 USPQ2d 1321, 1326 (Fed. Cir. 2005) (*en banc*). Further note *Brookhill-Wilk 1, LLC v. Intuitive Surgical, Inc.*, 334 F.3d 1294, 1298 67 USPQ2d 1132, 1136 (Fed. Cir. 2003) ("In the absence of an express intent to impart a novel meaning to the claim terms, the words are presumed to take on the ordinary and customary meanings attributed to them by those of ordinary skill in the art."). Accordingly, the PTO is not at liberty to change the ordinary and customary meaning of the claim term "instruction set" to be simply any computer instruction as that is not the ordinary and customary meaning of the term "instruction set" to those of ordinary skill in the art.

In addition, it is noted that not all ".dll" files and ".exe" files "each include a same instruction set" as the language of base independent Claims 1 and 6 requires. See the highlighted portion of the section "Looking At Code Generation" from page 495 of chapter 6

of the Microsoft C text by Jamsa (copy of page 495 also attached) that demonstrates that “EXE” type files do not always have the same instruction sets.

Consequently, as none of the references relied on teaches or suggests the requirement of base independent Claims 1 and 6 that the plurality of individual programs being compressed must “each include a same instruction set,” the rejection of Claim 1 (and Claim 4 dependent on Claim 1) and Claim 6 (and Claim 19 dependent on Claim 6) under 35 U.S.C. §103(a) as being unpatentable over Belu ‘762 in view of Videcrantz is clearly deficient, even if the “Winzip” screenshot documents are considered therewith. Therefore, withdrawal of this improper rejection of Claims 1, 4, 6, and 19 under 35 U.S.C. §103(a) as being unpatentable over Belu ‘762 in view of Videcrantz (with or without consideration of the “Winzip” screenshot documents) that ignores or misunderstands the ordinary and customary meaning of the claim term “instruction set” is respectfully submitted to be in order for this reason alone.

In addition, it is noted that this rejection of Claims 1, 4, 6, and 19 under 35 U.S.C. §103(a) as being unpatentable over Belu ‘762 in view of Videcrantz is essentially repeated from the Action mailed June 14, 2007, with the addition of the above-noted improper reliance on the “Winzip” screenshot documents and the misinterpretation of the claimed requirement for the same “instruction set” as referring to any and all computer instructions in general. This previously stated rejection was noted to be clearly improper in the response filed September 13, 2007, and in the Pre-Appeal Brief Conference Request filed November 14, 2007, because it violated precedent as to considering “means-plus-function” claims as set forth by the PTO reviewing court in *Gechter v. Davidson*, 116 F.3d 1454, 1460, 43 USPQ2d 1030, 1035 (Fed. Cir. 1997).

The *Gechter* decision could not be clearer as to requiring the PTO to construe the scope of the structures disclosed in the specification for the claimed “means,” to explain how

the “means” as disclosed in the specification are structurally equivalent to the relied upon elements embodied in the references, to define the exact function of each claimed “means,” and to explain how the relied upon reference elements are disclosed to perform the identical functions.

The outstanding Action once again fails to follow the above-noted *Gechter* required analysis as it fails, for example, to properly construe the independent Claim 1 “first generation means” (in the required manner of determining the scope of the structures disclosed in the specification for this claimed “means”) and it fails to explain how the “means” as disclosed in the specification are structurally equivalent to the relied upon elements embodied in the relied upon references (Belu ‘762 and Videcrantz) with or without consideration of the “Winzip” screenshot documents. Similarly missing is the required definition of the **exact** function of this “first generation means” that must **generate and output** the Claim 1 “first auxiliary data including a total number of individual programs combined and compressed by the compression means and a size of each individual program combined and compressed by the compression means.” Further missing is the required explanation as to “how the relied upon reference elements are disclosed to perform the identical function.”

With regard to the relied upon header teachings of paragraphs [0040]-[0042] of Belu ‘762 (noted at the bottom of page 2 of the outstanding Action), independent Claim 1 does not recite “having a file header that contains auxiliary data such as the names, sizes, compressed sizes, etc.” as urged at the bottom of page 2 of the outstanding Action. Also, the “Winzip” screenshot documents merely teaches a screenshot provided by a program, not any of the particular steps of that program, much less the interaction of the “Winzip” program with any particular compression system or the manner that the relied on “total 161 files” indication is produced. Further missing is the reason that the artisan would incorporate the cryptic

indication of “total 161 files” into the archive header portion 304 with the other information noted in paragraph [0057] as this paragraph of Belu ‘762 that teaches that “[t]he archive header portion 304 includes information about the entire archive that is to be created.” Furthermore, Belu ‘762 already teaches a “Zip” embodiment with a “Zip” header at paragraph [0044]. These “Zip” teachings raise the question of what would have led the artisan to adopt the “Winzip” screenshot documents teaching to replace these “Zip” teachings. In this regard, the Supreme Court, recently emphasized that “there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.” *KSR Int’l v. Teleflex Inc.*, 127 S.Ct. 1727, 82 USPQ.2d 1385, 1396 (2007) (quoting *In re Kahn*, 441 F.3d 977, 988, 78 USPQ2d 1329, 1336 (Fed. Cir. 2006). The required “articulated reasoning with some rational underpinning” (emphasis added) is clearly missing here.

Further as to the teachings of paragraphs [0040]-[0042] of Belu ‘762, a file header does not generate or output anything by itself and is clearly not the “equivalent” of the CPU disclosed in the specification (at page 11, lines 18-24) to count “the number of programs” and to calculate “the size of each of the programs.” Just as clearly, the “Winzip” screenshot documents simply establish when “Winzip 8.0” was shipped and present the screenshot from page 2 enlarged on page 1 and hand labeled “Figure 1” with further hand labeling relative to an indication of “total 161 files.” Whatever can be said of the “Winzip” screenshot documents, it cannot be reasonably said that the enlarged screenshot labeled Figure 1 discloses how the display element “total 161 files” (hand labeled 2) was determined, what the partially obliterated indication of “?.708KB” (hand labeled 3) next to it means, much less how the “size” numbers in the rows with the file names were determined. Also missing is any “Winzip” screenshot documents teaching that the illustrated “total 161 files” is a total number of **compressed** files as Claim 1 recites.

In this last regard, the indication of “total 161 files” (hand labeled 2) is ambiguous as it does not state that a total of 161 files have been *compressed*. It is well established that such ambiguous showings subject to different interpretations cannot be relied upon. *See, e.g., In re Turlay*, 304 F.2d 893, 899, 134 USPQ 355, 360 (CCPA 1962). Further note the requirements for clarity and definiteness of disclosure that the controlling precedent requires. Note, for example, *In re Hughes*, 145 USPQ 467, 471 (CCPA 1965) and *In re Moreton*, 129 USPQ 227, 230 (CCPA 1961).

Apparently realizing that the relied on header of paragraphs [0040]-[0042] of Belu cannot be reasonably equated to the above-noted independent Claim 1 recited “first generation means” or its function of “generating and outputting first auxiliary data including a total number of individual programs combined and compressed by the compression means and a size of each individual program combined and compressed by the compression means,” for example, page 3 of the outstanding Action attempts to introduce improper reliance on the doctrine of inherency to makeup for these shortcomings in Belu ‘762 and the “Winzip” screenshot documents. In this last regard, lines 10-11 at page 3 of the outstanding Action first draws the general unsubstantiated conclusion that “Belu, Winzip, and Videcrantz inherently discloses the compression means and generation means.”

This conclusion is clearly unsubstantiated as to the “Winzip” snapshot documents and Videcrantz because the outstanding action does not point to even a scintilla of actual evidence in the record indicating a disclosure of compression means and generation means in these references that are equivalent to the presently disclosed compression means and generation means, much less does the outstanding Action explain how these unidentified “means” disclosed by the “Winzip” snapshot documents and Videcrantz can be said to perform the claimed functions of these “means” in violation of the above-noted *Gechter* decision. Also missing is the presentation of any convincing technical reasoning that can be said to support



this self serving conclusion of inherency as to the “Winzip” screenshot documents and Videcrantz. It is well established that the PTO has the burden of proof to establish that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art. See *Ex parte Levy*, 17 USPQ2d 1461, 1464 (Bd. Pat. App. & Inter. 1990). This has not been done.

Further in this last regard, only paragraph [0022] of Belu ‘762 is mentioned at page 3 of the outstanding Action as disclosing compressing multiple files into a single file called an archive with further mention here of the data characteristics of the files being “examined” to determine ordering of the files in a sorted file list. The outstanding Action follows this general observation by the unsupported assertion that it is “inherent” that these operations are “performed on a computer system.” In this regard, it is clear that the compressing of data can be done by logic circuitry as explained on the enclosed copy of page 9 of U.S. Patent No. 6,058,056, taken from the USPTO web cite, the pertinent portion repeated here as follows:

FIG. 2 is a more detailed schematic block diagram of one embodiment of the error detection circuit 18 of FIG. 1. The error detection circuit 18 includes three data compression circuits 100-104 that collectively compress the data signals D1-D4 received from the arrays 22-28, as will be explained in more detail below. The data signals D1-D4 each include a complementary signal portion designated by the overbar in FIG. 2, with these complementary portions being omitted in FIG. 1 for the sake of brevity. A detailed schematic of one embodiment of the data **compression circuit** 100 is shown in FIG. 3. The data compression circuits 102 and 104 are identical to the data **compression circuit** 100 and thus, for the sake of brevity, only the circuit 100 will be described in more detail with reference to FIG. 3. The data **compression circuit** 100 includes a NAND gate 200 receiving the data signals D1 and D2 on its inputs, and a NAND gate 202 receiving the data signals D1 and D2 on its inputs. The output of the NAND gate 200 is applied to a gate of an NMOS drive transistor 204 and to a gate of a PMOS drive transistor 208. In response to the output of the NAND gate 200, the transistors 204 and 208 operate in a complementary manner to develop an output signal D[1-2] on a node 222. A first enable transistor 220 couples the source of the transistor 208 to a supply voltage source V<sub>CC</sub> in response to the test signal TEST applied to its gate through an inverter 206. A second enable transistor 216 couples the source of the transistor 204 to ground in response to the test signal TEST. The output of the NAND gate 202 is similarly coupled to a gate of an NMOS drive transistor 210 and to a gate of a PMOS drive transistor 214. In response to the output of the NAND gate 202, the transistors 210 and 214 operate in a

complementary manner to develop an output signal D[1-2] on an output node 224. The source of the transistor 210 is coupled through a third enable transistor 218 to ground in response to the test signal TEST, and the source of the transistor 214 coupled to the supply voltage source V<sub>CC</sub> through the enable transistor 220.

Not only can compression circuitry be implemented by logic gates, it is clear that recognizing “.dll” and “.exe” file extensions can be done by an operator viewing a listing of files on a computer display. The fact that a computer has an operating system does not mean that this operating system must select the files by detecting the file extensions as apparently erroneously argued at the bottom of page 3 of the outstanding Action.

Accordingly, the assertion in the outstanding Action that the teachings of paragraph [0022] of Belu ‘762 require implementation on a computer system is clearly mistaken as to an implementation using compression circuitry and there is no absolute certainty that a computer system must be used, much less how it would be used. See MPEP § 2112IV as follows:

The fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic. *In re Rijckaert*, 9 F.3d 1531, 1534, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993) (reversed rejection because inherency was based on what would result due to optimization of conditions, not what was necessarily present in the prior art); *In re Oelrich*, 666 F.2d 578, 581-82, 212 USPQ 323, 326 (CCPA 1981). "To establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.' " *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999) (citations omitted).

Also, the assertion at page 3 of the outstanding Action that “a computer inherently contains at least a processor, memory, and hard drive” is without merit as it contradicts the binding precedent of the PTO reviewing court set forth in *In re Paulsen*, 31 USPQ2d 1671, 1674 (Fed. Cir. 1994) as to the uncertain contents of a “computer” as follows:

The term “computer” is not associated with any one fixed or rigid meaning, as confirmed by the fact that it is subject to numerous definitions and

is used to describe a variety of devices with varying degrees of sophistication and complexity.

The court in *Paulsen* further noted that the term “computer” includes at least a calculator because this interpretation (by the USPTO) “was supported by authoritative lexicographic sources that confirmed that a calculator is considered to be a particular type of computer by those of ordinary skill in the art.” Clearly, a calculator device without “at least a processor, memory, and hard drive” can be considered to be a computer.

To whatever extent that Belu ‘762 teaches “compression” as to some combined files, as further noted at the bottom of page 3 of the outstanding Action, there is still no teaching in Belu ‘762 that this compression must be done by a computer or that the subject matter compressed is the Claim 1 subject matter. Also missing from any Belu ‘762 teaching as to “generating and outputting” is a teaching of something that must perform, inherently or otherwise, as the Claim 1 “first generation means” that must generate and output “first auxiliary data including a total number of individual programs combined and compressed by the compression means and a size of each individual program combined and compressed by the compression means.”

Thus, the rationale offered in the outstanding Action once again ignores the fact that it is the “first generation means” that must perform the recited function and the outstanding Action again errs in failing to supply any evidence in support of the conclusion of the asserted obviousness of this claimed “first generation means” or its claimed function. As previously noted, this lack of any evidence to support this conclusion and the other conclusions presented by the outstanding Action violates the Administrative Procedure Act that requires the PTO to provide “substantial evidence,” not mere unsubstantiated opinions and conclusions. See *In re Lee*, 217 F.3d 1365, 61 USPQ2d 1430, 1433-34 (Fed. Cir. 2002) as cited in the last response that notes:

Tribunals of the PTO are governed by the Administrative Procedure Act, and their rulings receive the same judicial deference as do tribunals of other administrative agencies. *Dickinson v. Zurko*, 527 U.S. 150, 50 USPQ2d 1930 (1999). Thus on appeal we review a PTO Board's findings and conclusions in accordance with the following criteria:

5 U.S.C. §706(2) The reviewing court shall—

(2) hold unlawful and set aside agency actions, findings, and conclusions found to be—

(A) arbitrary, capricious, an abuse of discretion, or otherwise not in accordance with law;

\* \* \* \*

(E) unsupported by substantial evidence in a case subject to sections 556 and 557 of this title or otherwise reviewed on the record of an agency hearing provided by statute;

For judicial review to be meaningfully achieved within these strictures, the agency tribunal must present a full and reasoned explanation of its decision. The agency tribunal must set forth its findings and the grounds thereof, as supported by the agency record, and explain its application of the law to the found facts. The Court has often explained:

The Administrative Procedure Act, which governs the proceedings of administrative agencies and related judicial review, establishes a scheme of “reasoned decision making.” Not only must an agency's decreed result be within the scope of its lawful authority, but the process by which it reaches that result must be logical and rational.

Not only does the rationale offered as to the use of assumptions and speculation in place of evidence violate the precedent established by the above noted *Lee* decision, it violates *In re Warner*, 379 F.2d 1011, 1017, 154 USPQ 173, 178 (CCPA 1967) (“The Patent Office has the initial duty of supplying the factual basis for its rejection. It may not, because it may doubt that the invention is patentable, resort to speculation, unfounded assumptions or hindsight reconstruction to supply deficiencies in its factual basis.”).

The relied upon teachings of Videcrantz do nothing to correct the above noted deficiencies of Belu ‘762. The addition of the “Winzip” screenshot documents also does not correct these Belu ‘762 shortcomings. Accordingly, the rejection of independent Claim 1 as being unpatentable over Belu ‘762 in view of Videcrantz under 35 U.S.C. §103(a) is clearly improper and should be withdrawn.

With further regard to independent method Claim 6, this independent claim recites, for example, a step of “generating first auxiliary data including a total number of individual programs that have been combined and compressed and a size of each individual combined and compressed program.” Just as the relied upon Belu file header is not a means for generating it also is not a method step of generating.

Furthermore, it is noted that Claim 6 is similar to Claim 1 as it requires the generating of auxiliary data that must indicate both the “total number of individual programs that have been combined and compressed and a size of each individual combined and compressed program.” As noted above, mere undocumented conclusions as to the ambiguous showing of the “Winzip” screenshot documents somehow teaching this claimed feature are also insufficient to demonstrate obviousness under 35 U.S.C. § 103.

Once again, the teachings of Videcrantz, or those of the “Winzip” screenshot documents, do nothing to correct the above noted deficiencies in Belu ‘762. Accordingly, the rejection of independent Claim 6 as being unpatenable over Belu ‘762 in view of Videcrantz under 35 U.S.C. §103(a), even considering the “Winzip” screenshot documents, is clearly improper and should be withdrawn.

Turning to dependent Claim 4, this is again a “means-plus-function” claim that has not been properly treated relative to the above-noted *Gechter v. Davidson* decision as no attempt has been made to construe the scope of the structures disclosed in the specification for the claimed “means,” no attempt has been made to explain how the “means” as disclosed in the specification are structurally equivalent to the relied upon portions of Belu ‘762, and the requirements as to addressing the claimed functions are again ignored. The inherency argument at the top of page 4 of the outstanding Action is once again based on speculation not any requirement in Belu ‘762 that compressed files are to be stored with auxiliary data in any thing. Also, Claim 4 requires the recited “storage means” to be “operatively connected

with the encryption means and the second generation means for storing said encrypted data received from the encryption means and said second auxiliary data received from said second generation means,” specifically claimed features not addressed by the outstanding Action that once again improperly relies on inherency at the top of page 4. However, nothing as to the header disclosure of paragraphs [0040]-[0042] of Belu ‘762 necessitates any storage, and certainly does not necessitate storage in a “RAM, a hard drive, or cache,” as improperly concluded at page 4, lines 2-3 of the outstanding Action.

Dependent method Claim 19 is similar to Claim 4 in that it requires “generating second auxiliary data indicating a size of said compressed program data” and “storing said encrypted data and said second auxiliary data,” which express claim features are again not adequately addressed by the outstanding Action, nor are these express claim features demonstrated to be inherent in the things or steps taught by Belu ‘762.

Consequently, as the teachings of Videcrantz or those of the “Winzip” screenshot document again do nothing to correct the above noted deficiencies of Belu ‘762, the rejection of dependent Claims 4 and 19 as being unpatentable over Belu ‘762 in view of Videcrantz under 35 U.S.C. §103(a), even considering the “Winzip” screenshot documents, is clearly improper for this reason as well. Accordingly, withdrawal of this improper rejection of dependent Claims 4 and 19 as being unpatentable over Belu ‘762 in view of Videcrantz under 35 U.S.C. §103(a) is further respectfully requested on this ground as well.

Turning to the rejection of independent Claims 7 and 11 under 35 U.S.C. §103(a) as being unpatentable over Belu ‘792 in view of Videcrantz and the “Winzip” screenshot documents, it is noted that Belu ‘792 is like Belu ‘762, Videcrantz and the “Winzip” screenshot documents in terms of failing to teach the requirement of base independent Claims 7 and 11 that the plurality of individual programs being compressed must “each include a same instruction set.”

Accordingly, this of Claims 7 and 11 under 35 U.S.C. §103(a) as being unpatentable over Belu '792 in view of Videcrantz and the "Winzip" screenshot documents is clearly deficient. Therefore, withdrawal of this improper rejection of independent Claims 7 and 11 under 35 U.S.C. §103(a) as being unpatentable over Belu '792 in view of Videcrantz and the "Winzip" screenshot documents that ignores or misunderstands the ordinary and customary meaning of the claim term "instruction set" is respectfully submitted to be in order for this reason alone.

In addition, it is noted that independent Claim 7 is a "means-plus-function" claim that has not been properly construed as noted above relative to the *Gechter* decision. In this regard, the outstanding Action again makes no attempt to construe the scope of the structures disclosed in the specification for the Claim 7 "means," that include a "creation means" and a "memory means" and their respective functions that were not present in Claims 1 or 4. The outstanding Action further violates the dictates of the *Gechter* decision because there is no attempt to explain how the "creation means" and "memory means" and other "means" of Claim 7 as disclosed in the specification are structurally equivalent to the relied upon portions of Belu '792. Similarly, the outstanding Action lacks any attempt to define the **exact** function of each of the claimed "means," and to explain how the relied upon reference elements are disclosed to perform the **identical** function.

Page 4 of the outstanding Action not only fails to set forth the required elements of a proper *Gechter* means and functions analysis, it misstates the limitations of Claim 7 and the teachings of Belu '792. With respect to the limitations of Claim 7, this claim actually recites that a "plurality of encrypted combined and compressed individual programs that each include a same instruction set" (emphasis added to subject matter omitted from the outstanding Action analysis) must be present with the "first auxiliary data indicating a total number of the individual programs that were combined and compressed and a size of each of

the combined and compressed individual programs” (emphasis again added to omitted subject matter). As was noted above, and now repeated as to Belu ‘792, as none of the references relied on teaches or suggests

To whatever extent paragraph [0088] of Belu ‘792 teaches decompression, it is decompression of the data portion 206 that was compressed from input file 202, a single file, not a file of combined programs. See paragraph [0059] of Belu ‘792 specifying compression of the single input file 202 to produce the compressed input file data portion 206. To whatever extent that paragraph [0066] of Belu ‘792 teaches providing the archive header portion 208 with the name of the (single) input file 202, the type of the (single) input file, the size of the compressed (single) input file, the size of the uncompressed (single) input file, etc., this auxiliary data has no relationship to the Claim 7 subject matter that must include “first auxiliary data indicating the total number of combined and compressed individual programs (not a single program), as well as the “size of each of the combined and compressed individual programs” not a size of one compressed program.

Even assuming that the artisan could provide encryption as taught by Videcrantz of the compressed single file 202 of Belu ‘792, this file 202 remains a single file with no relationship to the Claim 7 required “plurality of encrypted combined and compressed individual programs that each include a same instruction set and encrypted first auxiliary data indicating a total number of the individual programs that were combined and compressed and a size of each of the combined and compressed individual programs.”

Moreover, page 5 of the outstanding Action admits that neither Videcrantz nor Belu ‘792 teach or suggest the Claim 7 subject matter of:

creation means operatively connected to the decryption means for receiving the decrypted first auxiliary data and for creating a management table about locations of individual ones of said plurality of individual programs based on said decrypted first auxiliary data; and



memory means operatively connected to receive the decompressed and combined individual programs from the decompression means and to receive the management table from the creation means and for storing the decompressed and combined individual programs and said management table.

To overcome this admitted deficiency, the last paragraph on page 5 of the outstanding Action turns to the “Winzip” screenshot documents and an assertion that they disclose “a management table (figure 1) of all compressed files.” Clearly, however, the screenshot that is labeled as figure 1 on page 1 of these documents provides no information “about locations of individual ones of said plurality of individual programs based on said decrypted first auxiliary data” (emphasis added). Whether or not each compressed file listed on the screenshot “maps to a location within the compressed file” (emphasis added) as urged in the last paragraph on page 6, this does not convert the screenshot into the claimed management table having the claimed location information “based on said decrypted first auxiliary data” (emphasis added).

Furthermore, the rationale as to the screenshot of the “Winzip” screenshot documents being a generated management table the same as that of Claim 7 fails to explain why the artisan would store this screenshot supposedly giving information about plural files in a RAM or hard drive assumed to be part of the Belu ‘792 system (actually improperly assumed into existence by the continued misuse of the doctrine of inherency) when the Belu ‘792 system only compresses a single input file 202 to produce the compressed input file data portion 206, particularly as the information as to the input file 202 already is included in header portion 208 as noted in paragraph [0066] of Belu ‘792. As noted above, any suggested modification has to be accompanied by “articulated reasoning with some rational underpinning to support the legal conclusion of obviousness” as required by the above-noted *KSR Int’l* Supreme Court decision.

Furthermore, there is nothing that is inherent as to the Claim 7 “memory means” being a RAM or a hard drive because this is an improper assumption and not inherency as

noted above. Moreover, there is nothing taught or suggested in any of the applied references that would teach or suggest that any such RAM would also “receive the decompressed and combined individual programs from the decompression means” as there is only one file portion 206 as noted above.

Therefore, as the compression-encryption/decryption-decompression teachings of Videcrantz again do nothing to correct the above noted deficiencies in Belu ‘792, the rejection of independent Claim 7 as being unpatenable over Belu ‘792 in view of Videcrantz and the “Winzip” screenshot documents under 35 U.S.C. §103(a) is clearly improper and should be withdrawn.

Independent Claim 11 is similar to independent Claim 7 in presenting the functions of independent Claim 7 in method step form. The reasons why the rejection applied to Claim 11 is improperly based upon assumptions lacking evidence in support thereof and an improper use of inherency are, thus, the same as noted above as to independent Claim 7.

Once again, as the teachings of Videcrantz do nothing to correct the above noted deficiencies in Belu ‘792 and the “Winzip” screenshot documents, the rejection of independent Claim 11 as being unpatenable over Belu ‘792 in view of Videcrantz and the “Winzip” screenshot documents under 35 U.S.C. §103(a) is clearly improper and should be withdrawn.

The rejection of independent Claims 12 and 13 “under the same reasoning as claims 1 and 7” is traversed for all the reasons noted above as to the rejections of Claims 1 and 7. In addition, it is clear that base independent Claim 12 requires additional means relative to those recited by Claims 1 and 7. For example, Claim 12 requires at least the following means that are not recited by Claims 1 and 7:

second generation means for generating second auxiliary data  
indicating a size of said compressed program data;

storage means operatively connected to the encryption means and the second generation means for storing said encrypted data received from said encryption means and said second auxiliary data received from said second generation means;

selection means operatively connected with the decompression means for selecting a predetermined one of the plurality of combined individual programs from said plurality of combined individual programs received as said decompressed program data from said decompression means; and

execution means for receiving and executing said predetermined one of the plurality of combined individual programs.

As noted above, the outstanding Action is in error in failing to address all of these added means limitations and the associated functions of at least independent Claim 12 and those added by dependent Claim 13 as required by the above noted *Gechter* decision. In this regard, page 6 of the outstanding Action completely ignores all of these added “means” of Claim 12 that are not in Claims 1 and 7 except for the “execution means.” Even this limitation is misinterpreted at page 6 of the outstanding Action by referring to the “Winzip screen documents as showing such “executing means” when no such disclosure is present there. Also Belu ‘792 only executes the above-noted single file transformed into the universal self-executing file, it has nothing to do with receiving and executing a “predetermined one of the plurality of combined individual programs” as required by the function recited for the extracting means of Claim 12. Similarly, Videcrantz has no such teaching.

Thus, the rejection of independent Claim 12 and dependent Claim 13 as being “under the same reasoning as claims 1 and 7” and for the added reasons stated on page 6 of the outstanding Action is traversed and withdrawal of this clearly improper rejection is respectfully requested.

The rejection of Claims 13-15, 18, and 20 under 35 U.S.C. §103(a) as being unpatentable over Belu ‘762 in view of Videcrantz, “Winzip” screenshot documents, and Belu ‘792 is again traversed because each of these claims require that the plurality of

individual programs being compressed must “each include a same instruction set,” which is subject matter not taught or suggested by any of Belu ‘762, Belu ‘792, Videcrantz and the “Winzip” screenshot documents. Accordingly, the withdrawal of the rejection of Claims 13-15, 18, and 20 under 35 U.S.C. §103(a) as being unpatentable over Belu ‘762 in view of Videcrantz, “Winzip” screenshot documents, and Belu ‘792 is respectfully submitted to be in order for this reason alone.

Furthermore, Claims 13-15 ultimately depend from Claim 12 and the rejection here fails to adequately come to grips with all of the “means” limitations and associated functions of Claim 12 and misinterprets the teachings of the references and the actual function of the Claim 12 “executing means” as discussed above. Also the various “means” recitals of Claims 13-15 and their associated functions are not properly addressed as required by the above noted *Gechter* decision. In addition paragraph [0071] of Belu 792 mentioned at the bottom of page 6 of the outstanding rejection only teaches plural individual self executing files can be formed, not combining a plurality of files before compression and then compressing these plural combined files all sharing a same instruction set as required by parent independent Claim 12. The mischaracterization of the “Winzip” screenshot as being a “management table” at the bottom of page 6 of the outstanding Action was already treated above as was the failure of the outstanding Action to present any reasonable bases to store this screenshot as to the Claim 13 recited function of the Claim 13 recited “memory means” discussed above.

Accordingly, it is respectfully submitted that the rejection of Claims 13-15 under 35 U.S.C. §103(a) as being unpatentable over Belu ‘762 in view of Videcrantz, the “Winzip” screenshot documents, and Belu ‘792 is clearly in error for all of the above-noted reasons and withdrawal thereof is respectfully requested.

The rejection of independent Claim 18 under 35 U.S.C. §103(a) as being unpatentable over Belu '762 in view of Videcrantz, the "Winzip" screenshot documents, and Belu '792 is traversed for essentially the reasons noted above as to independent Claim 6 and dependent Claim 19. In this regard, independent Claim 18 is a method claim that essentially incorporates the limitations of independent Claim 6 and dependent Claim 19 along with some of the limitations of independent Claim 11. Accordingly, the rejection of independent Claim 18 under 35 U.S.C. §103(a) as being unpatentable over Belu '762 in view of Videcrantz, the "Winzip" screenshot documents, and Belu '792 is clearly improper and should be withdrawn.

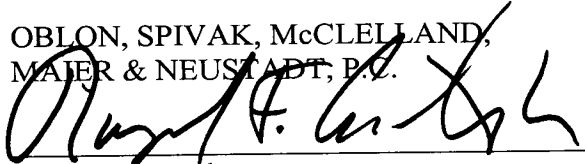
As Claim 20 depends directly on independent Claim 18, the rejection applied thereto under 35 U.S.C. §103(a) as being unpatentable over Belu '762 in view of Videcrantz, the "Winzip" screenshot documents, and Belu '792 is clearly improper and should be withdrawn for the reasons noted above as to this independent parent Claim 18. In addition, Claim 20 adds further features that are also not taught or suggested by Belu '762 in view of Videcrantz, the "Winzip" screenshot documents, and Belu '792 considered alone or together in any proper combination. Accordingly, the rejection applied thereto as being unpatentable over Belu '762 in view of Videcrantz, the "Winzip" screenshot documents, and Belu '792 under 35 U.S.C. §103(a) is clearly improper and should be withdrawn for this reason as well.

Application No. 10/743,088  
Reply to Office Action of 03/17/2008

As no further issues are believed to remain outstanding in the present application, it is believed that this application is clearly in condition for formal allowance and an early and favorable action to that effect is, therefore, respectfully requested.

Respectfully submitted,

OBLON, SPIVAK, McCLELLAND,  
MAIER & NEUSTADT, P.C.



Bradley D. Lytle  
Attorney of Record  
Registration No. 40,073

Customer Number  
**22850**

Tel: (703) 413-3000  
Fax: (703) 413 -2220  
(OSMMN 06/04)

Raymond F. Cardillo, Jr.  
Registration No. 40,440

I:\ATTY\RFC\24\247046-RECONSIDERATION.DOC

# MICROSOFT<sup>®</sup>

Secrets, Shortcuts, and Solutions

# C



K R I S J A M S A

PUBLISHED BY

Microsoft Press

A Division of Microsoft Corporation

16011 NE 36th Way, Box 97017

Redmond, Washington 98073-9717

Copyright © 1989 by Kris Jamsa

All rights reserved. No part of the contents of this book  
may be reproduced or transmitted in any form or by any means  
without the written permission of the publisher.

Library of Congress Cataloging in Publication Data

Jamsa, Kris A., 1960-

Microsoft C : secrets, shortcuts, and solutions.

p. cm.

Includes index.

1. C (Computer program language) 2. Microsoft C (Computer  
program) I. Title.

QA76.73.C15J355 1989 005.13'3--dc20 89-3430

ISBN 1-55615-203-5

CIP

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 FGFG 3 2 1 0 9

Distributed to the book trade in the United States by Harper & Row.

Distributed to the book trade in Canada by General Publishing Company, Ltd.

Distributed to the book trade outside the United States and Canada by Penguin Books Ltd.

Penguin Books Ltd., Harmondsworth, Middlesex, England

Penguin Books Australia Ltd., Ringwood, Victoria, Australia

Penguin Books N.Z. Ltd., 182-190 Wairau Road, Auckland 10, New Zealand

British Cataloging in Publication Data available

IBM®, PC/AT®, and PS/2® are registered trademarks  
of International Business Machines Corporation.

Microsoft®, MS-DOS®, and QuickC® are registered trademarks  
of Microsoft Corporation.

---

**Acquisitions Editor:** Dean Holmes

**Project Editor:** Eric Stroo

**Technical Editor:** Mike Halvorson

---



The following table lists the four warning levels. The higher the warning level, the more exhaustive the list of warning messages.

Level	Option	Meaning
0	/W0	Displays no warning messages
1	/W1	Displays most warning messages (the default)
2	/W2	Displays messages that indicate a return statement in type <i>void</i> functions or data conversions
3	/W3	Displays all warning messages

By default, CL uses /W1 to display warning messages. If you are developing an application for commercial use, you need the most exhaustive list of warnings you can obtain.

■ **TIP:** Use /W3 for production programs. In so doing, you might catch errors during compilation that would be difficult to debug.

## LOOKING AT CODE GENERATION

Every microprocessor has its own collection of instructions, called an instruction set. As microprocessors have increased in capability, so too have their instruction sets. The Intel 80286 microprocessor (found in the IBM PC AT) uses the 80286 instruction set, which is a superset of the 8086 instruction set used by the Intel 8086 microprocessor found in the IBM PC. This relationship permits the IBM PC AT to be compatible with the IBM PC. By default, the Microsoft C Compiler generates object code based on the 8086 instruction set. Compiled programs can therefore run on the 8086, 80186, 80286, and 80386 processors. If you are writing a program for an 80286 that does not need to run on an 8086, you can direct the C compiler to produce object code that maximizes performance by using the 80286 instruction set.

■ **FACT:** The /G0, /G1, or /G2 option directs CL to generate code for the 8086, 80186, or 80286 instruction set. The default compiler output is restricted to the 8086 instruction set, which you can specify explicitly with /G0.

Remember, once you generate code for the 80286, you decrease the portability of the executable file. In other words, you can no longer assume that the resulting EXE file can execute on a computer that uses an 8086 processor. The Microsoft C Compiler version 5.1 does not support code generation specifically for the 80386 instruction set.

■ **FACT:** The /Gc option directs CL to handle a function's arguments from left to right.

As discussed in Chapter 23, the C compiler passes arguments on the stack from right to left. Other programming languages, such as Pascal and FORTRAN, pass arguments on the stack from left to right. If you are writing a function to be called by other programming languages, you can use the /Gc option to direct CL to handle arguments from left to right.

develops a test clock signal TSTCLK having a frequency greater than the frequency of the clock signal CLK. The test clock signal TSTCLK is applied through a transfer gate 38 to the control circuit 16. The clock circuit 14 further includes a transfer gate 44 coupled between the clock terminal 34 and the control circuit 16 and receiving the test mode signal TM on its control input.

The test circuit 12 further includes an error detection circuit 18 receiving data signals D1-D4 from the arrays 22-28, respectively, and receiving control signals TEST, CLEAR, and ENABLE from the control circuit 16. In response to these control signals, the error detection circuit 18 compares the binary values of the data signals D1-D4, and develops an error signal ERROR on a terminal 46 indicating the result of this comparison, as will be explained in more detail below. The error detection circuit 18 can compare the binary values of the data signals D1-D4 to each other or to predetermined binary values, as understood by one skilled in the art.

FIG. 2 is a more detailed schematic block diagram of one embodiment of the error detection circuit 18 of FIG. 1. The error detection circuit 18 includes three data compression circuits 100-104 that collectively compress the data signals D1-D4 received from the arrays 22-28, as will be explained in more detail below. The data signals D1-D4 each include a complementary signal portion designated by the overbar in FIG. 2, with these complementary portions being omitted in FIG. 1 for the sake of brevity. A detailed schematic of one embodiment of the data **compression circuit** 100 is shown in FIG. 3. The data compression circuits 102 and 104 are identical to the data **compression circuit** 100 and thus, for the sake of brevity, only the circuit 100 will be described in more detail with reference to FIG. 3. The data **compression circuit** 100 includes a NAND gate 200 receiving the data signals D1 and D2 on its inputs, and a NAND gate 202 receiving the data signals D1 and D2 on its inputs. The output of the NAND gate 200 is applied to a gate of an NMOS drive transistor 204 and to a gate of a PMOS drive transistor 208. In response to the output of the NAND gate 200, the transistors 204 and 208 operate in a complementary manner to develop an output signal D[1-2] on a node 222. A first enable transistor 220 couples the source of the transistor 208 to a supply voltage source V.sub.CC in response to the test signal TEST applied to its gate through an inverter 206. A second enable transistor 216 couples the source of the transistor 204 to ground in response to the test signal TEST. The output of the NAND gate 202 is similarly coupled to a gate of an NMOS drive transistor 210 and to a gate of a PMOS drive transistor 214. In response to the output of the NAND gate 202, the transistors 210 and 214 operate in a complementary manner to develop an output signal D[1-2] on an output node 224. The source of the transistor 210 is coupled through a third enable transistor 218 to ground in response to the test signal TEST, and the source of the transistor 214 coupled to the supply voltage source V.sub.CC through the enable transistor 220.

In operation, the data **compression circuit** 100 operates in an active mode and an inactive mode in response to the test signal TEST. When the test signal TEST is inactive low, the enable transistors 216, 218 and 220 turn OFF isolating the drive transistors 204, 208, 210, and 214 from the supply voltage V.sub.CC and the ground. In this mode, high impedances are presented, respectively, on the output nodes 222 and 224 independent of the outputs of the NAND gates 200 and 202. When the test signal TEST is active high, the enable transistors 216 and 218 turn ON coupling the sources of the output transistors 204 and 210, respectively, to ground, and the enable transistor 220 turns ON coupling the sources of the output transistors 208 and 214 to the supply voltage source V.sub.CC. In the active mode, the state of the output signals D[1-2] and D[1-2] is determined by the binary values of the data signals D1, D1 and D2, D2. For example, assume the data signals D1 and D2 are both high. In response to the high data signals D1 and D2, the NAND gate 200 drives its output low turning OFF the transistor 204 and turning ON the transistor 208 which drives the voltage on the output node 222 high to approximately the supply voltage V.sub.CC through the transistors 208 and 220. When the data signals D1 and D2 are high, the data signals D1 and D2 are accordingly low. In response to the low data signals D1 and D2, the NAND gate 202 drives its output high, turning OFF the transistor 214 and turning ON

and system architecture 130.

One aspect of the invention is that the processor supports multiple system architectures. Thus, the number of instruction sets and/or system architectures supported, as well as the type of instruction sets and system architectures supported, are not critical to this aspect of the invention. What is important to this aspect of the invention is that the processor can switch between the instruction set architectures and system architectures. For example, alternative embodiments may support one instruction set and two system architectures. As another example, alternative embodiments may support three instruction set architectures and two system architectures. Other alternative embodiments may support three instruction set architectures and three system architectures. An embodiment which supports two instruction set architectures and two system architectures is described so as not to obscure the invention.

FIG. 2 shows a functional block diagram illustrating the selectable configurations or modes of a processor according to one embodiment of the invention. FIG. 2 shows a line 200 representing that the processor includes an instruction set unit 203 and a system unit 207. FIG. 2 also shows that instruction set unit 203 selectively operates in either an instruction set configuration 210 or in an instruction set configuration 220. In one embodiment, instruction set configuration 210 includes segmentation unit 215. Segmentation unit 215 allows for compatibility with existing x86 memory management techniques which utilize segmentation. In addition, FIG. 2 shows system unit 207, which selectively operates in either a system configuration 230 or a system configuration 240.

Instruction set unit 203 executes instructions from a first instruction set while instruction set configuration 210 is selected. In one embodiment, this first instruction set is based on the 16/32-bit **x86 instruction set** used by existing Intel microprocessors. This instruction set operates using what are referred to as effective or logical addresses. Instruction set configuration 210 sends these effective addresses to segmentation unit 215 which translates them into linear addresses. The technique of segmentation is well known in the prior art and is further described in the following reference: Shanley, Tom and Anderson, Don, ISA System Configuration, MindShare, Inc. (1993). Thus, instruction set configuration 210 with segmentation unit 215 provides a first instruction set architecture. Alternative embodiments which support other instruction sets may require other address translation techniques (rather than or in addition to segmentation), or may not require any address translation.

Instruction set unit 203 executes instructions from a second instruction set which is different from the first instruction set, while instruction set configuration 220 is selected. In one embodiment, this second instruction set is a 64-bit instruction set which operates using the same format of address generated by segmentation unit 215 (i.e., linear addresses). Since this 64-bit instruction set uses linear addresses, it can address the entire 64-bit virtual address space and does not require segmentation. In this manner, instruction set configuration 220 provides a second instruction set architecture.

Thus, instruction set unit 203 includes all necessary software, firmware, and hardware to provide for the execution of two instruction sets. In one embodiment, instruction set unit 203 includes at least one prefetch unit, decode unit, and execution unit, as well as a mechanism for switching between the two instruction set configurations (not shown). One embodiment of instruction set unit 203 will be later described with reference to FIG. 8. While one embodiment of instruction set unit 203 has been described in which it is implemented on the processor, alternative embodiments could implement all or part of instruction set unit 203 in hardware residing outside the processor, or in software.

System unit 207 provides a first system architecture while system configuration 230 is selected. This first system architecture supports typical operating system functions according to a first system technique. In one embodiment, system configuration 230 is compatible with existing x86 processors and includes an event handling unit 233 and a paging unit 236. Event handling unit 233 provides for the